

Modeling Social Annotation: a Bayesian Approach

Anon Plangprasopchok and Kristina Lerman
USC Information Sciences Institute

Collaborative tagging systems, such as *Delicious*, *CiteULike*, and others, allow users to annotate resources, e.g., Web pages or scientific papers, with descriptive labels called tags. The social annotations contributed by thousands of users, can potentially be used to infer categorical knowledge, classify documents or recommend new relevant information. Traditional text inference methods do not make best use of social annotation, since they do not take into account variations in individual users' perspectives and vocabulary. In a previous work, we introduced a simple probabilistic model that takes interests of individual annotators into account in order to find hidden topics of annotated resources. Unfortunately, that approach had one major shortcoming: the number of topics and interests must be specified *a priori*. To address this drawback, we extend the model to a fully Bayesian framework, which offers a way to automatically estimate these numbers. In particular, the model allows the number of interests and topics to change as suggested by the structure of the data. We evaluate the proposed model in detail on the synthetic and real-world data by comparing its performance to Latent Dirichlet Allocation on the topic extraction task. For the latter evaluation, we apply the model to infer topics of Web resources from social annotations obtained from *Delicious* in order to discover new resources similar to a specified one. Our empirical results demonstrate that the proposed model is a promising method for exploiting social knowledge contained in user-generated annotations.

Categories and Subject Descriptors: H.2.8 [DATABASE MANAGEMENT]: Database Applications—*Data mining*; I.5.1 [PATTERN RECOGNITION]: Models—*Statistical*

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Collaborative Tagging, Probabilistic Model, Resource Discovery, Social Annotation, Social Information Processing

1. INTRODUCTION

A new generation of Social Web sites, such as *Delicious*, *Flickr*, *CiteULike*, *YouTube*, and others, allow users to share content and annotate it with metadata in the form of comments, notes, ratings, and descriptive labels known as tags. Social annotation captures the collective knowledge of thousands of users and can potentially be used to enhance a number of applications including Web search, information personalization and recommendation, and even synthesize categorical knowledge [Schmitz 2006; Mika 2007]. In order to make best use of user-generated metadata, we need methods that effectively deal with the challenges of data sparseness and noise, as well as take into account variations in the vocabulary, interests, and the level of expertise among individual users.

Consider specifically *tagging*, which has become a popular method for annotating con-

Author's address: USC Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292.

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 2010 ACM 0000-0000/2010/0000-0001 \$5.00

ACM Journal Name, Vol. x, No. y, zz 2010, Pages 1–0??.

tent on the Social Web. When a user tags a resource, be it a Web page on the social bookmarking site *Delicious*, a scientific paper on *CiteULike*, or an image on the social photosharing site *Flickr*, the user is free to select any keyword, or tag, from an uncontrolled personal vocabulary to describe the resource. We can use tags to categorize resources, similar to the way documents are categorized using their text, although the usual problems of sparseness (few unique keywords per document), synonymy (different keywords may have the same meaning), and ambiguity (same keyword has multiple meanings), will also be present in this domain. Dimensionality reduction techniques such as topic modeling [Hofmann 1999; Blei et al. 2003; Buntine et al. 2004], which project documents from word space to a dense topic space, can alleviate these problems to a certain degree. Specifically, such projections address the sparseness and synonymy challenges by combining “similar” words together in a topic. Similarly, the challenge of word ambiguity in a document is addressed by taking into account the senses of co-appearing words in that document. In other words, the sense of the word is determined jointly along with the other words in that document.

Straightforward application of the previously mentioned methods to social annotation would aggregate resource’s tags over all users, thereby losing important information about individual variation in tag usage, which can actually help the categorization task. Specifically, in social annotation, similar tags may have different meanings according to annotators’ perspectives on the resource they are annotating [Lerman et al. 2007]. For example, if one searches for Web resources about car prices using the tag “jaguar” on *Delicious*, one receives back a list of resources containing documents about luxury cars and dealers, as well as guitar manuals, wildlife videos, and documents about Apple Computer’s operating system. The above mentioned methods would simply compute tag occurrences from annotations across all users, effectively treating all annotations as if they were coming from a single user. As a result, a resource annotated with the tag “jaguar” will be undesirably associated with any sense of the keyword simply based on the number of times that keyword (tag) was used for each sense.

We claim that users express their individual interests and vocabulary through tags, and that we can use this information to learn a better topic model of tagged resources. For instance, we are likely to discover that users who are interested in luxury cars use the keyword “jaguar” to tag car-related Web pages; while, those who are interested in wildlife use “jaguar” to tag wildlife-related Web pages. The additional information about user interests is essential, especially since social annotations are generally very sparse.¹ In a previous work, [Plangprasopchok and Lerman 2007], we proposed a probabilistic model that takes into account interest variation among users to infer a more accurate topic model of tagged resources. In this paper we describe a Bayesian version of the model (Section 3). We explore its performance in detail on the synthetic data (Section 4.1) and compare it to Latent Dirichlet Allocation (LDA) [Blei et al. 2003], a popular document modeling algorithm. We show that in domains with high tag ambiguity, variations among users can actually help discriminate between tag senses, leading to a better topic model. Our model is, therefore, best suited to make sense of social metadata, since this domain is characterized both by a high degree of noise and ambiguity and a highly diverse user population with varied

¹There are only 3.74 tags on average for a certain photo in Flickr [Rattenbury et al. 2007]. In addition, there are 4 to 7 tags used by each user on a certain URL from our observation in *Delicious* data we obtained; while tag vocabulary on an resource gets stable after few bookmarks as reported in [Golder and Huberman 2006].

interests.

As a second contribution of the paper, we incorporate a Hierarchical Dirichlet Process [Teh et al. 2006] to create an adaptive version of the proposed model (Section 5), which enables the learning method to automatically adjust the model parameters. This capability helps overcome one of the main difficulties of applying the original model to the data: namely, having to specify the right number of topics and interests.

Finally, the proposed models are validated on a real-world data set obtained from the social bookmarking site *Delicious* (Section 4.2 and Section 5.2). We first train the model on this data, then measure the quality of the learned topic model. Specifically, the learned topic model is used as a compressed description of each Web resource. We compute similarity between resources based on the compressed description and manually evaluate results to show that the topic model obtained by the method proposed in this paper identifies more similar resources than the baseline.

2. MODELING SOCIAL ANNOTATION

In general, a social annotation system involves three entities: resources (e.g., Web pages on *Delicious*), users and metadata. Although there are different forms of metadata, such as descriptions, notes and ratings, we focus on tags only in this context. We define a variable R as resources, U as users, and T as tags. Their realizations are defined as r , u and t respectively. A post (or bookmark) k on a resource r by a user u , can be formalized as a tuple $\langle r, u, \{t_1, t_2, \dots, t_j\} \rangle_k$, which can be further broken down into co-occurrence of j resource-user-tag triples: $\langle r, u, t \rangle$. N_R , N_U and N_T are the number of distinct resources, users and tags respectively.

In addition to the observable variables defined above, we introduce two ‘hidden’ or ‘latent’ variables, which we will attempt to infer from the observed data. The first variable, Z , represents resource topics, which we view as categories or concepts of resources. From our previous example, the tag “jaguar” can be associated with topics ‘cars’, ‘animals’, ‘South America’, ‘computers’, etc. The second variable, X , represents user interests, the degree to which users subscribe to these concepts. One user may be interested in collecting information about luxury cars before purchasing one, while another user may be interested in vintage cars. A user u has her interest profile, ψ_u , which is a weight distribution over all possible interests x . And ψ (without subscript) is just an $N_U \times N_X$ matrix. Similarly, a resource r has its topic profile, ϕ_r , which is again a weight distribution over all possible topics z , whereas ϕ (without subscript) is an $N_R \times N_Z$ matrix. Thus, a resource about South American jaguars will have a higher weight on ‘animals’ and ‘South America’ topics than on the ‘cars’ topic. Usage of tags for a certain interest-topic pair (x, z) is defined as a weight distribution over tags, $\theta_{x,z}$ – that is, some tags are more likely to occur for a given pair than others. The weight distribution of all tags, θ , can be viewed as an $N_T \times N_Z \times N_X$ matrix.

We cast an annotation event as a stochastic process as follows:

- User u finds a resource r interesting and would like to bookmark it.
- For each tag that u generates for r :
 - User u selects an interest x from her interest profile ψ_u ; resource r selects a topic z from its topic profile ϕ_r .
 - Tag t is then chosen based on users’s interest and resource’s topic from the interest-topic distribution over all tags $\theta_{x,z}$.

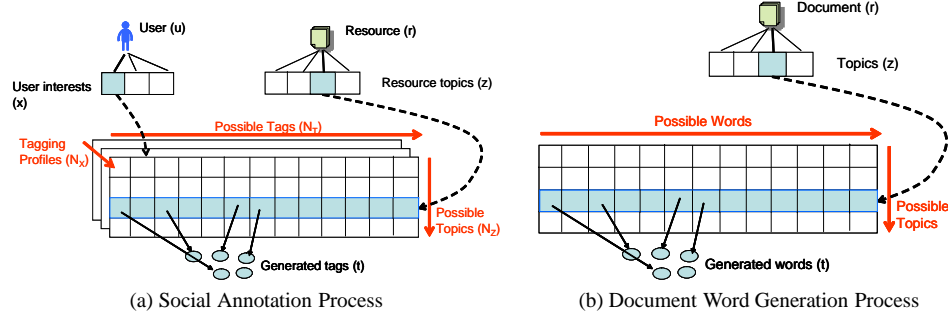


Fig. 1. Schematic diagrams represent: (a) tag generation process in social annotation domain; (b) word generation process in document modeling domain.

This process is depicted schematically in Figure 1(a). Specifically, a user u has an interest profile, represented by a vector of interests ψ_u . Meanwhile, a resource r has its own topic profile, represented by a vector of topics ϕ_r . Users who share the same interest (x) have the same tagging policy — the tagging profile “plate”, shown in the diagram. For the “plate” corresponding to an interest x , each row corresponds to a particular topic z , and it gives $\theta_{x,z}$, the distribution over all tags for that topic and interest.

The process can be compared to the word generation process in standard topic modeling approaches, e.g., LDA [Blei et al. 2003] and pLSA [Hofmann 2001], as shown in Figure 1 (b). In topic modeling, words of a certain document are generated according to a single policy, which assumes that all authors of documents in the corpus share the same tagging patterns. In other words, a set of “similar” tags is used to represent a topic across all authors. In our “jaguar” example, for instance, we may find one topic to be strongly associated with words “cars”, “automotive”, “parts”, “jag”, etc., while another topic may be associated with words “animals”, “cats”, “cute”, “black”, etc., and still another with “guitar”, “fender”, “music”, etc. and so on.

In social annotation, however, a resource can be annotated by many users, who may have different opinions, even on the same topic. Users who are interested in restoring vintage cars will have a different tagging profile than those who are interested in shopping for a luxury car. The ‘cars’ topic would then decompose under different tagging profiles into one that is highly associated with words “restoration”, “classic”, “parts”, “catalog”, etc., and another that is associated with words “luxury”, “design”, “performance”, “brand”, etc. The separation of tagging profiles for each group of users who share the same interest provides a machinery to address this issue and constitutes the major distinction between our work and standard topic modeling.

3. FINITE INTEREST TOPIC MODEL

In our previous work [Plangprasopchok and Lerman 2007], we proposed a probabilistic model that describes social annotation process, which was extended from probabilistic Latent Semantic Analysis (pLSA) [Hofmann 2001]. However, the model inherited some shortcomings from pLSA. First, the strategy for estimating parameters in both models — the point estimation using EM algorithm — has been criticized as being prone to local maxima [Griffiths and Steyvers 2004; Steyvers and Griffiths 2006]. In addition, there

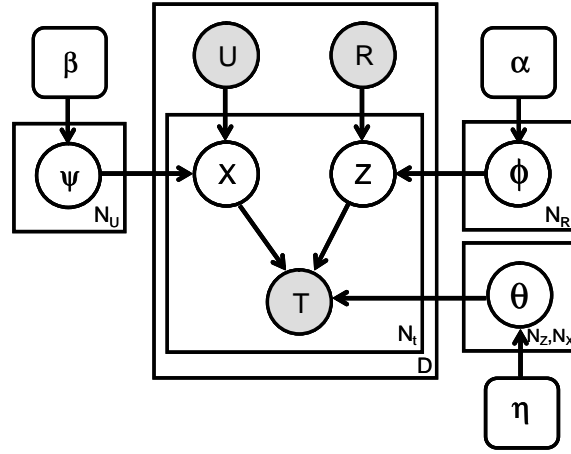


Fig. 2. Graphical representation of the social annotation process. R, U, T, X and Z denote variables “Resource”, “User”, “Tag”, “Interest” and “Topic” respectively. ψ, ϕ and θ are distributions of user over interests, resource over topics and interest-topic over tags respectively. N_t represents the number of tag occurrences for one bookmark (by a particular user, u , on a particular resource, r); D represents the number of all bookmarks in the social annotation system. The hyperparameters α, β , and η control dispersions of categorical topics, interests and tags respectively.

is also no explicit way to extend these models to automatically infer the dimensions of parameters, such as the number of components used to represent topics (N_Z) and interests (N_X).

We extend our previous Interest Topic Model (ITM) the same way pLSA was upgraded to Latent Dirichlet Allocation (LDA) model [Blei et al. 2003]. In other words, we implement the model under a Bayesian framework, which offers solutions [Blei et al. 2003; Griffiths and Steyvers 2004; Neal 2000] to the previously mentioned problems. By doing so, we introduce priors on top of parameters ψ, ϕ and θ to make the model fully generative, i.e., the mechanism for generating these parameters is explicitly implemented. To make the model analytically simple, we use symmetric Dirichlet priors. Following the generative process described in Section 2, the model can be described as a stochastic process, depicted a graphical form [Buntine 1994] in Figure 2:

- . $\psi_u \sim \text{Dirichlet}(\beta/N_X, \dots, \beta/N_X)$ (generating user u interest’s profile)
- . $\phi_r \sim \text{Dirichlet}(\alpha/N_Z, \dots, \alpha/N_Z)$ (generating resource r topic’s profile)
- . $\theta_{x,z} \sim \text{Dirichlet}(\eta/N_T, \dots, \eta/N_T)$. (generating tag’s profile for interest x and topic z)

For each tag t_i of a bookmark,

- . $x_i \sim \text{Discrete}(\psi_u)$
- . $z_i \sim \text{Discrete}(\phi_r)$
- . $t_i \sim \text{Discrete}(\theta_{x_i, z_i})$.

One possible way to estimate parameters is to use Gibbs sampling [Gilks et al. 1996; Neal 2000]. Briefly, the idea behind the Gibbs sampling is to iteratively use the parameters

of the current state to estimate parameters of the next state. In particular, each next-state parameter is sampled from the posterior distribution of that parameter given all other parameters in the previous state. The sampling process is done sequentially until sampled parameters approach the target posterior distributions. Recently, this approach was demonstrated to be simple to implement, yet competitively efficient, and to yield relatively good performance on the topic extraction task [Griffiths and Steyvers 2004; Rosen-Zvi et al. 2004].

Since we use Dirichlet priors, it is straightforward to integrate out ψ , ϕ and θ . Thus, we only need to sample hidden variables \mathbf{x} and \mathbf{z} and later on estimate ψ , ϕ and θ once \mathbf{x} and \mathbf{z} approach their target posterior distribution. To derive Gibbs sampling formula for sampling \mathbf{x} and \mathbf{z} , we first assume that all bookmarks are broken into N_K tuples. Each tuple is indexed by i and we refer to the observable variables, resource, user and tag, of the tuple i as r_i , u_i , t_i . We refer to the hidden variables, topic and interest, for this tuple as z_i and x_i respectively, with \mathbf{x} and \mathbf{z} representing the vector of interests and topics over all tuples.

We define $N_{r_i, z_{-i}}$ as the number of all tuples having $r = r_i$ and z but excluding the present tuple i . In words, if $z = z_i$, $N_{r_i, z_{-i}} = N_{r_i, z_i} - 1$; otherwise, $N_{r_i, z_{-i}} = N_{r_i, z_i}$. Similarly, N_{z_{-i}, x_i, t_i} is a number of all tuples having $x = x_i$, $t = t_i$ and z but excluding the present tuple i ; \mathbf{z}_{-i} represents all topic assignments except that of the tuple i . The Gibbs sampling formulas for sampling z and x , whose derivation we provide in the Appendix, are as follows.

$$p(z_i | \mathbf{z}_{-i}, \mathbf{x}, \mathbf{t}) = \frac{N_{r_i, z_{-i}} + \alpha/N_Z}{N_{r_i} + \alpha - 1} \cdot \frac{N_{z_{-i}, x_i, t_i} + \eta/N_T}{N_{z_{-i}, x_i} + \eta} \quad (1)$$

$$p(x_i | \mathbf{x}_{-i}, \mathbf{z}, \mathbf{t}) = \frac{N_{u_i, x_{-i}} + \beta/N_X}{N_{u_i} + \beta - 1} \cdot \frac{N_{x_{-i}, z_i, t_i} + \eta/N_T}{N_{x_{-i}, z_i} + \eta} \quad (2)$$

Consider Eq. (1), which computes a probability of a certain topic for the present tuple. This equation is composed of 2 factors. Suppose that we are currently determining the probability that the topic of the present tuple i is j ($z_i = j$). The left factor determines the probability of topic j to which the resource r_i belongs according to the present topic distribution of r_i . Meanwhile, the right factor determines the probability of tag t_i under the topic j of the users who have interest x_i . If resource r_i assigned to the topic j has many tags, and the present tag t_i is “very important” to the topic j according to the users with interest x_i , there is a higher chance that tuple i will be assigned to topic j . A similar insight is also applicable to Eq. (2). In particular, suppose that we are currently determining the probability that the interest of the present tuple i is k ($x_i = k$). If user u_i assigned to the interest k has many tags, and tag t_i is “very important” to the topic z_i according to users with interest k , the tuple i will be assigned to interest k with higher probability.

In the model training process, we sample topic z and interest x in the current iteration using their assignments from the previous iteration. By sampling z and x using Eq. (1) and Eq. (2) for each tuple, the posterior distribution of topics and interests is expected to converge to the true posterior distribution after enough iterations. Although it is difficult to assess convergence of Gibbs sampler in some cases as mentioned in [Sahu and Roberts 1999], we simply monitor it through the likelihood of data given the model, which measures how well the estimated parameters fit to the data. Once the likelihood reaches the

stable state, it only slightly fluctuates from one iteration to the next, i.e., there is no systematic and significant increase and decrease in likelihood. We can use this as a part of the stopping criterion. Specifically, we monitor likelihood changes over a number of consecutive iterations. If the average of these changes is less than some threshold, the estimation process terminates. More robust approaches to determining the stable state are discussed elsewhere, e.g. [Ritter and Tanner 1992]. The formula for the likelihood is defined as follows.

$$f(\mathbf{t}; \psi, \phi, \theta) = \prod_{i=1:N_K} \left(\frac{N_{x_i, z_i, t_i} + \eta/N_T}{N_{x_i, z_i} + \eta} \right) \quad (3)$$

To avoid a numerical precision problem in model implementation, one usually uses log likelihood $\log(f(\mathbf{t}; \psi, \phi, \theta))$ instead. Note that we use the strategy mentioned in [Escobar and West 1995] (Section 6) to estimate α , β and η from data.

The sampling results in the stable state are used to estimate model parameters. Again, we define $N_{r,z}$ as the number of all tuples associated with resource r and topic z , with N_r , $N_{x,u}$, N_u , $N_{x,u,t}$ and $N_{x,z}$ defined in a similar way. From Eq. (18) and Eq. (19) in the Appendix, the formulas for computing such parameters are as follows:

$$\phi_{r,z} = \frac{N_{r,z} + \alpha/N_Z}{N_r + \alpha} \quad (4)$$

$$\psi_{u,x} = \frac{N_{u,x} + \beta/N_X}{N_u + \beta} \quad (5)$$

$$\theta_{x,z,t} = \frac{N_{x,z,t} + \eta/N_T}{N_{x,z} + \eta} \quad (6)$$

Parameter estimation via Gibbs sampling is less prone to the local maxima problem than the generic EM algorithm, as argued in [Rosen-Zvi et al. 2004]. In particular, this scheme does not estimate parameters ϕ , ψ , and θ directly. Rather, they are integrated out, while the hidden variables z and x are iteratively sampled during the training process. The process estimates the “posterior distribution” over possible values of ϕ , ψ , and θ . At a stable state, z and x are drawn from this distribution and then used to estimate ϕ , ψ , and θ . Consequently, these parameters are estimated from a combination of “most probable solutions”, which are obtained from multiple maxima. This clearly differs from the generic EM with point estimation, which we used in our previous work [Plangprasopchok and Lerman 2007]. Specifically, the point estimation scheme estimates ϕ , ψ , and θ from single local maximum.

Per training iteration, the computational complexity of Gibbs sampling is more expensive than EM. This is because we need to sample hidden variables (z and x) for each data point (tuple), whereas EM only requires updating parameters. In general, the number of the data points is larger than the dimension of parameters. However, it has been reported in [Griffiths and Steyvers 2004] that to reach the same performance, Gibbs sampling requires fewer floating point operations than the other popular approaches: Variational Bayes and Expectation Propagation [Minka 2001]. Moreover, to our knowledge, there is currently no

explicit way to extend these approaches to automatically infer the size of hidden variables, as Gibbs sampling can. Note that inference of these numbers is described in Section 5.

4. EVALUATION

In this section we evaluate the Interest Topic Model and compare its performance to LDA [Blei et al. 2003] on both synthetic and real-world data. The synthetic data set enables us to control the degree of tag ambiguity and individual user variation, and examine in detail how both learning algorithms respond to these key challenges of learning from social metadata. The real-world data set, obtained from the social bookmarking site *Delicious*, demonstrates the utility of the proposed model.

The baseline in both comparisons is LDA, a probabilistic generative model originally developed for modeling text documents [Blei et al. 2003], and more recently extended to other domains, such as finding topics of scientific papers [Griffiths and Steyvers 2004], topic-author associations [Rosen-Zvi et al. 2004], user roles in a social network [McCallum et al. 2007], and Collaborative Filtering [Marlin 2004]. In this model, the distribution of a document over a set of topics is first sampled from a Dirichlet prior. For generating each word in the document, a topic is first sampled from the distribution; then, a word is selected from the distribution of topics over words. One can apply LDA to model how tags are generated for resources on social tagging systems. One straightforward approach is to ignore information about users, treating all tags as if they came from the same user. Then, a resource can be viewed as a document, while tags across different users who bookmarked it are treated as words, and LDA is then used to learn parameters.

ITM extends LDA by taking into account individual variations among users. In particular, a tag for a certain bookmark is chosen not only from the resource’s topics but also from user’s interests. This allows each user group (with the same interest) to have its own policy, $\theta_{x,z,t}$, for choosing tags to represent a topic. Each policy is then used to update resource topics as in Eq. (1). Consequently, $\phi_{r,z}$ is updated based on interests of users who actually annotated resource r , rather than updating it from a single policy that ignores user information. We thus expect ITM to perform better than LDA when annotations are made by diverse user groups, and especially when tags are ambiguous.

4.1 Synthetic Data

To verify the intuition about ITM, we evaluated the performance of the learning algorithms on synthetic data. Our data set consists of 40 resources, 10 topics, 100 users, 10 interests, and 100 tags. We first separate resources into five groups, with resources in each group assigned topic weights from the same (Dirichlet) probability distribution, which forces each resource to favor 2–4 out of ten topics. Rather than simulate the tagging behavior of user groups by generating individual tagging policy plates as in Figure 1(a), we simplify the generative process to simulate the impact of diversity in user interests on tagging. To this end, we represent user interests as distributions over topics.

We create data sets under different tag ambiguity and user interest variation levels. To make these settings *tunable*, we generate distributions of topics over tags, and distributions of resources over topics using symmetric Dirichlet distributions with different parameter values. Intuitively, when sampling from the symmetric Dirichlet distribution² with a low

²Samples that are sampled from Dirichlet distribution are discrete probability distributions

parameter value, for example 0.01, the sampled distribution contributes weights (probability values that are greater than zero) to only a few elements. In contrast, the distribution will contribute weights to many elements when it is sampled from a Dirichlet distribution with a high parameter value. We used this parameter of the symmetric Dirichlet distribution to adjust *user variation*, i.e., how broad or narrow user interests are, and *tag ambiguity*, i.e., how many or how few topics each tag belongs to. With higher parameter values, we can simulate a behavior of more ambiguous tags, such as “jaguar”, which has multiple senses, i.e., it has weights allocated to many topics. Low parameter values can be used to simulate low ambiguity tags, such as “mammal”, which has one or few senses. The parameter values used in the experiments are 1, 0.5, 0.1, 0.05 and 0.01.

To generate tags for each simulated data set, user interest profiles ψ_u are first drawn from the symmetric Dirichlet distribution with the same parameter value. A similar procedure is done for distributions of topics over words θ . A resource will presumably be annotated by a user if the match between resource’s topics and user’s interests is greater than some threshold. The match is given by the inner product between the resource’s topics and user’s interests, and we set the threshold at $1.5 \times$ the average match computed over all user-resource pairs. The rationale behind this choice of threshold is to ensure that a resource will be tagged by a user who is strongly interested in the topics of that resource. When the user-resource match is greater than threshold, a set of tags (a post or bookmark) is generated according to the following procedure. First, we compute the topic distribution from an element-wise product of the resource’s topics and user’s interests. Next, we sample a topic from this distribution and produce a tag from the tag distribution of that topic. This guarantees that tags are only generated according to user’s interests. We repeat this process seven times in each post³ and eliminate redundant tags. The process of generating tags is summarized below:

```

for each resource-user pair  $(u, r)$  do
   $m_{r,u} = \phi_r \cdot \psi_u$  (compute the match score)
end for
 $\bar{m} = \text{Average}(\mathbf{m})$ 
for each resource-user pair  $(r, u)$  do
  if  $m_{r,u} > 1.5\bar{m}$  then
     $\text{topicPref} = \phi_r \times \psi_u$  (element-wise product)
    for  $i = 1$  to 7 do
       $z \sim \text{topicPref}$  (draw a topic from the topic preference)
       $t_{r,u}^i \sim \theta_z$  (sample  $i^{\text{th}}$  tag for the  $(u, r)$  pair)
    end for
    Remove redundant tags
  end if
end for

```

We measure sensitivity to tag ambiguity and user interest variation for LDA and ITM on the synthetic data generated with different values of symmetric Dirichlet parameters. One way to measure sensitivity is to determine how the learned topic distribution, ϕ_r^{ITM} or ϕ_r^{LDA} , deviates from the actual topic distribution of resource r , ϕ_r^{Actual} . Unfortunately, we cannot compare them directly, since topic order of the learned topic distribution may

³We chose seven because *Delicious* users in general use four to seven tags in each post.

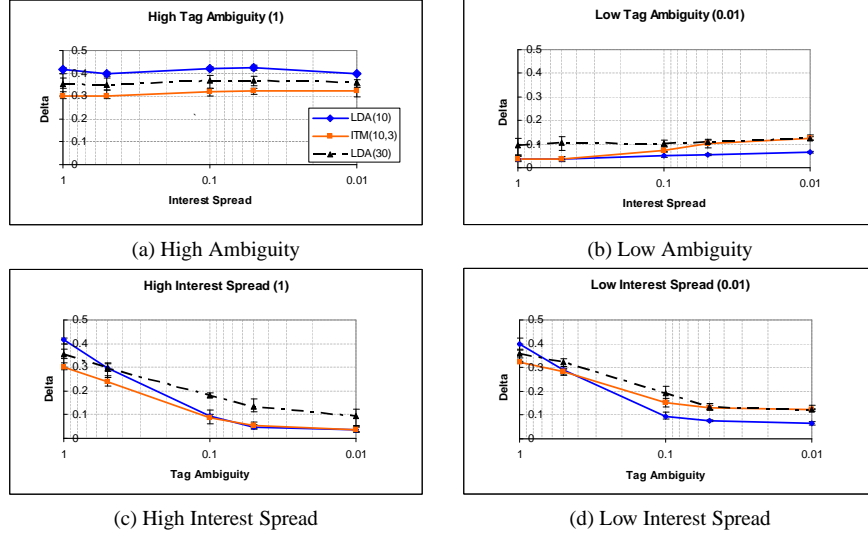


Fig. 3. Deviations, Δ , between actual and learned topics on synthetic data sets for different regimes: (a) high tag ambiguity; (b) low tag ambiguity; (c) high interest spread; (d) low interest spread. LDA(10) and LDA(30) refers to LDA that is trained with 10 and 30 topics respectively; ITM(10,3) refers to ITM that is trained with 10 topics and 3 interests.

not be the same as that of the actual one.⁴ An indirect way to measure this deviation is to compare distances between pairs of resources computed using the actual and learned topic distributions. We define this deviation as Δ . We calculate the distance between two distributions using Jensen-Shannon divergence (JSD) [Lin 1991]. If a model accurately learned the resources' topic distribution, the distance between two resources computed using the learned distribution will be equal to the distance computed from the actual distribution. Hence, the lower Δ , the better model performance. The deviation between the actual and learned topic distributions is

$$\Delta = \sum_{r=1}^{N_R} \sum_{r'=r+1}^{N_R} |JSD(\phi_r^{Learned}, \phi_{r'}^{Learned}) - JSD(\phi_r^{Actual}, \phi_{r'}^{Actual})|. \quad (7)$$

Δ is computed separately for each algorithm, $Learned = ITM$ and $Learned = LDA$.

We ran both LDA and ITM to learn distributions of resources over topics, ϕ , for simulated data set generated with different values of tag ambiguity and user interest variation. We set the number of topics to 10 for each model, and the number of interests to three for ITM. Both models were initialized with random topic and interest assignments and then trained using 1000 iterations. For the last 100 iterations, we used topic and interest assignments in each iteration to compute ϕ (using Eq. (4) for ITM and Eq. (7) in [Griffiths and Steyvers 2004] for LDA). The average⁵ of ϕ in this period is then used as the distributions

⁴This property of probabilistic topic models is called exchangeability of topics [Steyvers and Griffiths 2006].

⁵The reason to use the average of ϕ is that, in the stable state, the topic/interest assignments can still fluctuate from one iteration to another. To avoid estimate ϕ from an iteration that possibly has idiosyncratic topic/word assignments, one can average ϕ over multiple iterations [Steyvers and Griffiths 2006].

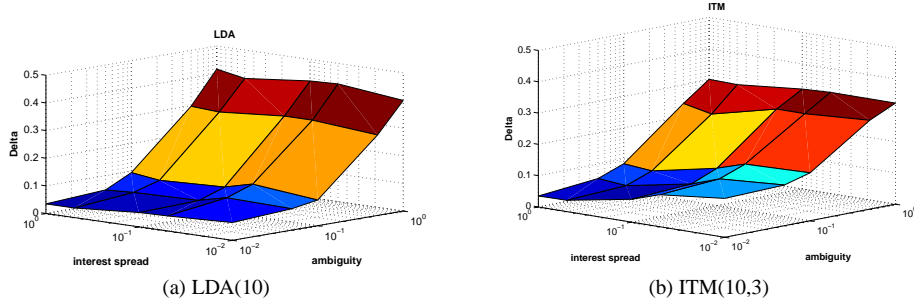


Fig. 4. This plot shows the deviation Δ between actual and learned topics on synthetic data sets, under different degrees of tag-ambiguity and user interest variation. The Δ of LDA is shown on the left (a); as that of ITM is on the right (b). The colors were automatically generated by the plotting program to improve readability.

of resources over topics. We ran the learning algorithm five times for each data set.

Deviations between learned topics and actual ones of simulated data sets are shown in Figure 3 and Figure 4. In the case when degree of tag ambiguity is high, ITM is superior to LDA for the entire range of user interest variation, as shown in Figure 3(a). This is because ITM exploits user information to help disambiguate tag senses; thus, it can learn better topics, which are closer to the actual ones, than LDA. In the other regime, when tag ambiguity is low, user information does not help and can even degrade ITM performance, especially when the degree of interest variation is low, as in Figure 3(b). This is because low amount of user interest variation demotes statistical strength of the learned topics. Suppose that, for example, we have two similar resources: the first one is bookmarked by one group, the second bookmarked by another. If these two groups have very different interest profiles, ITM will tend to split the “actual” topic that describes those resources into two different topics — one for each group. Hence, each of these resources will be assigned to a different learned topic, resulting in a higher Δ for ITM.

In the case when user interest variation is high (Figure 3(c)), ITM is superior to LDA for the same reason that it uses user information to disambiguate tag senses. Of course, there is no advantage to using ITM when the degree of tag ambiguity is very low, and it yields similar performance to LDA. In the last regime, when interest variation is low (Figure 3(d)), ITM is superior to LDA for high degree of tag ambiguity, even though its topics may lose some statistical strength. ITM’s performance starts to degrade when tag ambiguity degree is low, for the same reason as in Figure 3(b). These results are summarized in 3D plots in Figure 4.

We also ran LDA with 30 topics, in order to compare LDA to ITM, when both models have the same complexity. As shown in the Figure 3, with the same model complexity, ITM is preferable to LDA in all settings. In some cases, LDA with higher complexity (30 topics) is inferior to the LDA with lower complexity (10 topics). We suspect that this degradation is caused by over-specification of the model with too many topics.

For the computational complexity, both LDA and ITM are required to sample the hidden variables for all data points using Gibbs sampling. For LDA, only the topic variable z is needed to be sampled; for ITM, the interest variable x is also required. The computational cost in each sampling is proportional to a number of topics, N_Z , for z , and that of interest, N_X , for x . Let define κ as a constant. We also define a number of all datapoints (tuples)

as N_K . Hence, a computational cost for LDA, in each iteration can be approximated as $N_K \times (\kappa \times N_Z)$. The computational cost of ITM in each iteration can be approximated as $N_K \times (\kappa \times (N_Z + N_X))$.

In summary, ITM is not superior to LDA in learning topics associated with resources in every case. However, we showed that ITM is preferable to LDA in scenarios characterized by a high degree of tag ambiguity and some user interest variation, which is the case in the social annotations domain.

4.2 Real-World Data

In this section we validate the proposed model on real-world data obtained from the social bookmarking site *Delicious*. The hypothesis we make for evaluating the proposed approach is that the model that takes users into account can infer higher quality (more accurate) topics ϕ than those inferred by the model that ignores user information.

The “standard” measure⁶ used for evaluating topic models is the perplexity score [Blei et al. 2003; Rosen-Zvi et al. 2004]. Specifically, it measures generalization performance on how a certain model can predict unseen observations. In document topic modeling, a portion of words in each document are set aside as testing data; while the rest are used as training data. Then the perplexity score is computed from a conditional probability of the testing given training data. This evaluation is infeasible in the social annotation domain, where each bookmark contains relatively few tags, compared to document’s words.

Instead of using perplexity, we propose to directly measure the quality of the learned topics on a simplified resource discovery task. The task is defined as follows: “given a *seed* resource, find other most similar resources” [Ambite et al. 2009]. Each resource is represented as a distribution over learned topics, ϕ , which is computed using Eq. (4). Topics learned by the better approach will have more discriminative power for categorizing resources. When using such distribution to rank resources by similarity to the seed, we would expect the more similar resources to be ranked higher than less similar resources. Note that similarity between a pair of resources A and B is computed using Jensen-Shannon divergence (JSD) [Lin 1991] on their topic distributions, ϕ_A and ϕ_B .

To evaluate the approach, we collected data for five seeds: *flytecomm*,⁷ *geocoder*,⁸ *wunderground*,⁹ *whitepages*,¹⁰ and *online-reservationz*.¹¹ The *flytecomm* allows users to track flights given the airline and flight number or departure and arrival airports; *geocoder* returns geographic coordinates of a given address; *wunderground* gives weather information for a particular location (given by zipcode, city and state, or airport); *whitepages* returns person’s phone numbers and *online-reservationz* lists hotels available in some city on some dates. We crawl *Delicious* to gather resources possibly relating to each seed. The crawling strategy is as follows: for each seed

—retrieve the 20 most popular tags associated with this resource.

—For each of the tags, retrieve other resources that have been annotated with the tag.

⁶In fact, topic model’s evaluation is still currently in controversy according to a personal communication at <http://nlpers.blogspot.com/2008/06/evaluating-topic-models.html> by Hal Daumé.

⁷<http://www.flytecomm.com/cgi-bin/trackflight/>

⁸<http://geocoder.us>

⁹<http://www.wunderground.com/>

¹⁰<http://www.whitepages.com/>

¹¹<http://www.online-reservationz.com/>

Seed	# Resources	# Users	# Tags	#Tripples
Flytecomm	3,562	34,594	14,297	2,284,308
Geocoder	5,572	46,764	16,887	3,775,832
Wunderground	7,176	45,852	77,056	6,327,211
Whitepages	6,455	12,357	64,591	2,843,427
Online-Resevationz	764	41,003	9,194	162,763

Table I. The table presents statistics for five data sets for evaluating models' performance. Note that a triple is a resource, user, and tag co-occurrence.

—For each resource, collect all bookmarks (resource-user-tag triples).

We wrote a special-purpose page scraper to extract this information from *Delicious*. In principle, we could continue to expand the collection of resources by gathering tags and retrieving more resources tagged with those keywords, but in practice, even after a small traversal, we already obtain millions of triples. In each corpus, each resource has at least one tag in common with the seed. Statistics on these data sets are presented in Table I.

For each corpus, LDA is trained with 80 topics, while the number of topics and interests for ITM is set to 80 and 40 respectively. The topic and interest assignments are randomly initialized, and then both models are trained with the 500 iterations.¹² For the last 100 iterations, we use the topic and interest assignments, in each iteration, to compute the distributions of resources over topics, ϕ . The average of ϕ in this period is then used as the distributions of resources over topics.

Next, the learned distributions of resources over topics, ϕ , are used to compute the similarity of resources in each corpus to the seed. The performance of each model is evaluated by manually checking the 100 most similar resources produced by the model. A resource is judged to be similar if it provides an input form that takes semantically the same inputs as the seed and returns semantically the same data. Hence, *flightaware*¹³ is judged similar to *flytecomm* because both take flight information and return flight status.

Figure 5 shows the number of relevant resources identified within the top x resources returned by LDA and ITM. From the results, we can see that ITM is superior to LDA in three data sets: *flytecomm*, *geocoder* and *online-reservationz*. However, its performance for *wunderground* and *whitepages* is about the same as that of LDA. Although we have no empirical proof, we hypothesize that weather and directory services are of interest to all users, and are therefore bookmarked by a large variety of users, unlike users interested in tracking flights or booking hotels online. As a result, ITM cannot exploit individual user differences to learn more accurate topics ϕ in the *wunderground* and *whitepages* cases.

To illustrate utility of ITM, we select examples of topics and interests of the model induced from the *flytecomm* corpus. For purposes of visualization, we first list in descending order the top tags that are highly associated with each *topic*, which are obtained from θ_z (aggregated over all interests in the topic z). For each topic, we then enumerate some *interests*, and present a list of top tags for each interest, obtained from $\theta_{x,z}$. We manually label topics and interests (in *italics*) according to the meaning of its dominant tags.

Travel & Flights topic: travel, Travel, flights, airfare, airline, flight, airlines, guide, aviation, hotels, deals,

¹²We discovered that the model converging very quickly. In particular, the model appear to reach the stable state within 300 iterations in all data sets

¹³<http://flightaware.com/live/>

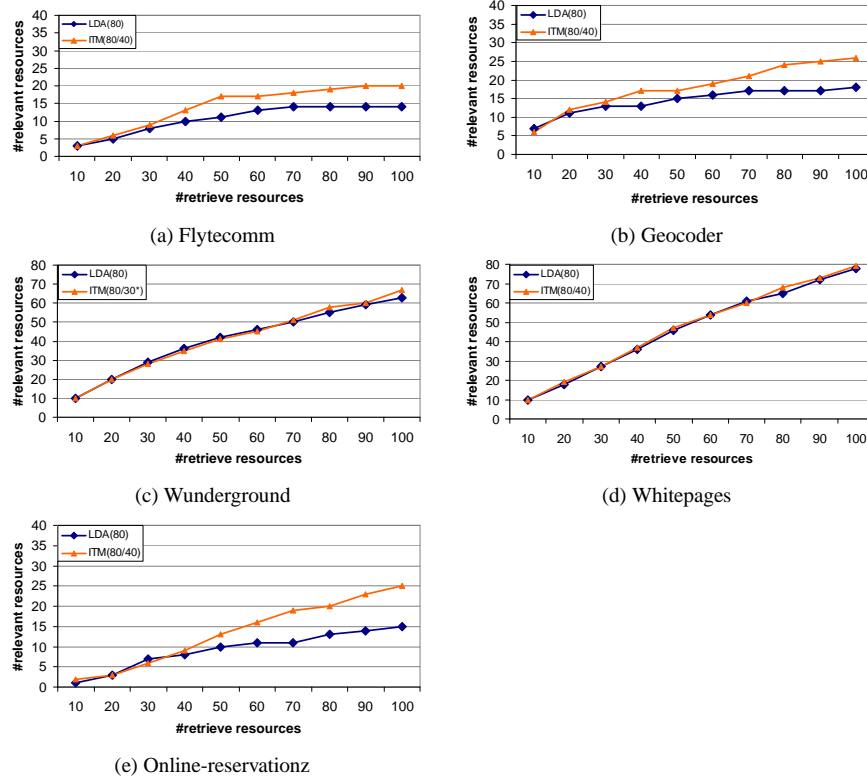


Fig. 5. Performance of different models on the five data sets. X-axis represents the number of retrieved resources; y-axis represents the number of relevant resources (that have the same function as the seed). LDA(80) refers to LDA that is trained with 80 topics. ITM(80/40) refers to ITM that is trained with 80 topics and 40 interests. In *wunderground* case, we can only run ITM with 30 interests due to the memory limits.

reference, airplane

—*Flight Tracking interest*: travel, flight, airline, airplane, tracking, guide, flights, hotel, aviation, tool, packing, plane

—*Deal & Booking interest*: vacation, service, travelling, hotels, search, deals, europe, portal, tourism, price, compare, old

—*Guide interest*: travel, cool, useful, reference, world, advice, holiday, international, vacation, guide, information, resource

Video & p2p topic: video, download, bittorrent, p2p, youtube, media, torrent, torrents, movies, videos, Video, downloads, dvd, free, movie

—*p2p Video interest*: video, download, bittorrent, youtube, torrents, p2p, torrent, videos, movies, dvd, media,

googlevideo, downloads, pvr

- Media & Creation interest*: video, media, movies, multimedia, videos, film, editing, vlog, remix, sharing, rip, ipod, television, videoblog
- Free Video interest*: video, free, useful, videos, cool, downloads, hack, media, utilities, tool, hacks, flash, audio, podcast

Reference topic: reference, database, cheatsheet, Reference, resources, documentation, list, links, sql, lists, resource, useful, mysql

- Databases interest*: reference, database, documentation, sql, info, databases, faq, technical, reviews, tech, oracle, manuals
- Tips & Productivity interest*: reference, useful, resources, information, tips, howto, geek, guide, info, productivity, daily, computers
- Manual & Reference interest*: resource, list, guide, resources, collection, help, directory, manual, index, portal, archive, bookmark

The three interests in the “Travel & Flights” topic have obviously different themes. The dominant one is more about tracking status of a flight; while the less dominant ones are about searching for travel deals and traveling guides respectively. This implies that there are subsets of users who have different perspectives (or what we call interests) towards the same topic. Similarly, different interests also appear in the following topics, “Video & p2p” and “Reference.”

Figure 6 presents examples of topic distributions for three resources learned by LDA and ITM: the seed *flytecomm*, *usatoday*,¹⁴ and *bookings*.¹⁵ Although all are about travel, the first two resources have specific flight tracking functionality; while the last one is about hotel & trip booking. In distribution of resources over the topics learned by LDA, shown in Figure 6 (a), all resources have high weights on topics #1 and #2, which are about traveling deals and general aviation. In the case of topics learned by ITM, shown in Figure 6 (b), *flytecomm* and *usatoday* have their high weight on topic #25, which is about tracking flights, while *bookings* does not. Consequently, ITM will be more helpful than LDA in identifying flight tracking resources. This demonstrates the advantage of ITM in exploiting individual differences to learn more accurate topics.

5. INFINITE INTEREST TOPIC MODEL

In Section 3, we assumed that parameters, such as, N_Z and N_X (number of topics and interests respectively), were fixed and known *a priori*. The choice of values for these parameters can conceivably affect the model performance. The traditional way to determine these numbers is to learn the model several times with different values of parameters, and then select those that yield the best performance [Griffiths and Steyvers 2004].

¹⁴<http://www.usatoday.com/travel/flights/delays/tracker-index.htm>

¹⁵<http://www.bookings.org/>

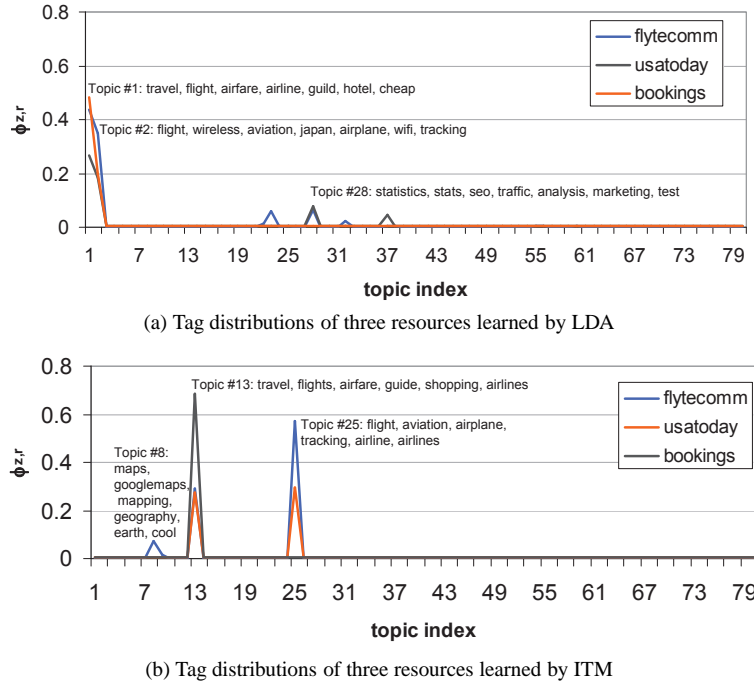


Fig. 6. Topic distributions of three resources: *flytocomm*, *usatoday*, *bookings* learned by (a) LDA and (b) ITM. $\phi_{z,r}$ in y-axis indicates a weight of the topic z in the resource r – the degree to which r is about the topic z .

In this work, we choose another solution by extending our finite model to have “countably” infinite numbers of topics and interests. By “countably” infinite number of components, we mean that such numbers are flexible and can vary according to the number of observations. Intuitively, there is a higher chance that more topics and interests will be found in a data set that has more resources and users. Such unbounded number of components can be dealt with within a Bayesian framework, as mentioned in the previous works [Neal 2000; Rasmussen 2000; Teh et al. 2006]. This approach helps bypass the problem of selecting values for these parameters.

Following [Neal 2000], we set both N_Z and N_X to approach ∞ . This will give the model the ability to select not only previously used topic/interest components but also to instantiate “unused” components when required. However, the model that we derived in the previous section cannot be extended directly under this framework due to the use of symmetric Dirichlet priors. As pointed out by [Teh et al. 2006], when the number of components grows, using the symmetric Dirichlet prior results in a very low — even zero probability — chance that a mixture component is shared across groups of data. That is, in our context, there is a higher chance that a certain topic is only used within one resource rather than utilized by many of them. Considering Eq. (1), if we set N_Z to approach ∞ , we can obtain posterior probability of z as follows

$$p(z_i = z_{used} | \mathbf{z}_{-i}, \mathbf{x}, \mathbf{t}) = \frac{N_{r_i, z_{-i}}}{N_{r_i} + \alpha - 1} \cdot \frac{N_{z_{-i}, x_i, t_i} + \eta / N_T}{N_{z_{-i}, x_i} + \eta} \quad (8)$$

$$p(z_i = z_{new} | \mathbf{z}_{-i}, \mathbf{x}, \mathbf{t}) = \frac{\alpha}{N_{r_i} + \alpha - 1} \cdot \frac{1}{N_T} \quad (9)$$

From Eq. (8), we can perceive that the model only favors topic components that are only used within the resource r_i . Meanwhile, for other components that are not used by that resource, N_{z_{-i}, x_i, t_i} would equal zero and thus result in zero probability in choosing them. Consequently, the model only chooses topic components for a resource either from components that are currently used by that resource, or it instantiates a new component for that resource with probabilities according to Eq. (8) and Eq. (9) respectively. As more new components are instantiated, each resource tends to own its components exclusively. From the previous section, we can also perceive that each resource profile is generated independently (using symmetric Dirichlet prior) — there is no mechanism to link the used components across different resources¹⁶. As mentioned in [Teh et al. 2006], this is an undesired characteristic, because, in our context, we would expect “similar” resources to be described by the same set of “similar” topics.

One possible way to handle this problem is to use Hierarchical Dirichlet Process (HDP) [Teh et al. 2006] as the prior instead of the symmetric Dirichlet prior. The idea of HDP is to link components at group-specific level together by introducing global components across all groups. Each group is only allowed to use some (or all) of these global components and thus, some of them are expected to be shared across several groups. We adapt this idea by considering all tags of resource r to belong to the resource group r . Similarly, all tags of user u belong to the user group u . Each of the resource groups is assigned to some topic components selected from the global topic component pool. Similarly, each of the user groups is assigned to some interest components selected from the global interest component pool. This extension is depicted in Figure 7. Suppose that a number of all possible topic components is N_Z (which will be set to approach ∞ later on) and that for interest components is N_X , we can describe such extension as a stochastic process as follows.

At the global level, the weight distribution of components is sampled according to

. $(\beta_1, \dots, \beta_{N_X}) \sim \text{Dirichlet}(\gamma_x / N_X, \dots, \gamma_x / N_X)$ (generating global interest component weight)

. $(\alpha_1, \dots, \alpha_{N_Z}) \sim \text{Dirichlet}(\gamma_z / N_Z, \dots, \gamma_z / N_Z)$ (generating global topic component weight)

where γ_x and γ_z are concentration parameter, which controls diversity of interests and topics at global level.

At the group specific level,

¹⁶This behavior can be easily observed in multiple samples, each drawn independently from a Dirichlet distribution $\text{Dirichlet}(\alpha_1, \dots, \alpha_k)$. If α_i is “small” and k is “large”, there is a higher chance that samples obtained from this Dirichlet distribution will have no overlapped component i.e., for any pair of samples, there is no case when the same components have their value greater than 0 at the same time. Lack of this component overlap across samples will be obvious when $k \rightarrow \infty$. This is the problem that can be found in the model with infinite limit on N_Z and N_X .

- $\psi_u \sim \text{Dirichlet}(\mu_x \cdot \beta_1, \dots, \mu_x \cdot \beta_{N_X})$ (generating user u interest's profile)
- $\phi_r \sim \text{Dirichlet}(\mu_z \cdot \alpha_1, \dots, \mu_z \cdot \alpha_{N_Z})$ (generating resource r topic's profile)

where μ_x and μ_z are concentration parameter, which controls diversity of interests and topics at group specific level. The remaining steps involving generation of tags for each bookmark are the same as in the previous process.

Suppose that there is an infinite number of all possible topics, $N_Z \rightarrow \infty$, and a portion of them are currently used in some resources. By following [Teh et al. 2006], we can rewrite the global weight distribution of topic components, α , as $(\alpha_1, \dots, \alpha_{k_z}, \alpha_u)$, where k_z is the number of currently used topics components and $\alpha_u = \sum_{k=k_z+1}^{N_Z} \alpha_k$ - all of unused topic components. Similarly, we can write $(\alpha_1, \dots, \alpha_{k_z}, \alpha_u) \sim \text{Dirichlet}(\gamma_z/N_Z, \dots, \gamma_z/N_Z, \gamma_{z_u})$, where $\gamma_z/N_Z = \gamma_z/N_Z$ and $\gamma_{z_u} = \frac{(N_Z - k_z) \cdot \gamma_z}{N_Z}$. The same treatment is also applied to that of interest components.

Now we can generalize Eq. (1) and Eq. (2) for sampling posterior probabilities of topic z and interest x with HDP priors as follows.

For sampling topic component assignment for datapoint i ,

$$p(z_i = k | \mathbf{z}_{-i}, \mathbf{x}, \mathbf{t}) = \frac{N_{r_i, z_{-i}} + \mu_z \alpha_k}{N_{r_i} + \mu_z - 1} \cdot \frac{N_{z_{-i}, x_i, t_i} + \eta/N_T}{N_{z_{-i}, x_i} + \eta} \quad (10)$$

$$p(z_i = k_{new} | \mathbf{z}_{-i}, \mathbf{x}, \mathbf{t}) = \frac{\mu_z \alpha_u}{N_{r_i} + \mu_z - 1} \cdot \frac{1}{N_T} \quad (11)$$

For sampling interest component assignment for datapoint i ,

$$p(x_i = j | \mathbf{x}_{-i}, \mathbf{z}, \mathbf{t}) = \frac{N_{u_i, x_{-i}} + \mu_x \beta_j}{N_{u_i} + \mu_x - 1} \cdot \frac{N_{x_{-i}, z_i, t_i} + \eta/N_T}{N_{x_{-i}, z_i} + \eta} \quad (12)$$

$$p(x_i = j_{new} | \mathbf{x}_{-i}, \mathbf{z}, \mathbf{t}) = \frac{\mu_x \beta_u / N_X}{N_{u_i} + \mu_x - 1} \cdot \frac{1}{N_T}, \quad (13)$$

where k and j are an index for topic and interest component respectively. From these equations, we allow the model to instantiate a new component from the pool of unused components. Considering the case when a new topic component is instantiated and, for simplicity, we set this new component to be the last used component, indexed with k'_z . We need to obtain weight $\alpha_{k'_z}$ for this new component and also update the weight of all unused components, α_u . From the unused component pool, we know that one of its unused components will be chosen as a newly used component, k'_z , with probability distribution $(\alpha_{k_z+1}/\alpha_u, \dots, \alpha_{N_Z}/\alpha_u)$ which can be sampled from $\text{Dirichlet}(\gamma_z/N_Z, \dots, \gamma_z/N_Z)$. Suppose the component k'_z will be chosen from one of these components and we collapse the remaining unused components. It will be chosen with the probability $\alpha_{k'_z}/\alpha_u$, which can be sampled from $\text{Beta}(\gamma_z/N_Z, \gamma_{z_u}/N_Z - \gamma_z/N_Z)$, where $\text{Beta}(\cdot)$ is a Beta distribution.

Now, suppose k'_z is chosen. The probability of choosing this component is updated to $\alpha_{k'_z}/\alpha_u \sim \text{Beta}(\gamma_z/N_Z + 1, \gamma_{z_u}/N_Z - \gamma_z/N_Z)$. When $N_Z \rightarrow \infty$, this reduces to $\alpha_{k'_z}/\alpha_u \sim \text{Beta}(1, \gamma_{z_u}/N_Z)$. Hence, to update $\alpha_{k'_z}$, we first draw $a \sim \text{Beta}(1, \alpha_u)$. We then update $\alpha_{k'_z} \leftarrow a \cdot \alpha_u$ and update $\alpha_{u'} \leftarrow (1 - a) \cdot \alpha_u$. Similar steps are also applied to interest components.

Note that if we compare Eq. (10) to Eq. (8), the problem we found so far has gone since $p(z_i = k | \mathbf{z}_{-i}, \mathbf{z}, \mathbf{t})$ will never have zero probability even if $N_{r_i, z-i} = 0$.

At the end of each iteration, we use the same method [Teh et al. 2006] to sample α and β and update hyperparameters $\gamma_z, \gamma_x, \mu_z, \mu_x$ using the method described in [Escobar and West 1995]. We refer to this infinite version of ITM as “Interest Topic Model with Hierarchical Dirichlet Process” (HDPITM) for the rest of the paper.

For the computational complexity, although N_Z and N_X are both set to approach ∞ , the computational cost of each iteration, however, does not approach ∞ . Considering Eq. (10) and Eq. (11), sampling of z_i only involves currently instantiated topics plus one “collapsed topic”, which represents all currently unused topics. Similarly, the sampling of x_i only involves currently instantiated interests plus one. For a particular iteration, a computational cost for HDP can therefore be approximated as $N_K \times (\kappa \times \bar{N}_Z + 1)$. And that for HDPITM can be approximated as $N_K \times (\kappa \times (\bar{N}_Z + \bar{N}_X + 2))$, where \bar{N}_Z and \bar{N}_X are respectively the average number of topics and interests in that iteration.

5.1 Performance on the synthetic data

We ran both HDP and HDPITM to extract topic distributions, ϕ , on the simulated data set. In each run the number of instantiated topics was initialized to ten, which equals to the actual number of topics for both HDP and HDPITM. The number of interests was initialized to three. Similar to the setting in Section 4.1, topic and interest assignments were randomly initialized and then trained using 1000 iterations. Subsequently, ϕ was computed from the last 100 iterations. The results are shown in Figure 8 (a) and (b) for HDP and HDPITM respectively. From these results, the behaviors of both model for different settings are somewhat similar to those of LDA and ITM. In particular, HDPITM can exploit user information to help disambiguate tag senses, while HDP cannot. Hence, the performance of HDPITM is better than that of HDP when tag ambiguity level is high. And since topics may lose some statistical strength under low user interest condition, HDPITM is inferior to HDP, similar to Figure 3(b) for the finite case.

As one can compare the plots (a) and (b) in Figure 3 and Figure 8, the performance of infinite model is generally worse than that of the finite one, even though we allow the former the ability to adjust topic/interest dimensions. One possible factor is that the model still allows topic/interest dimensions (configuration) to change even though the trained model is in a “stable” state. That would prohibit the model from optimizing its parameters for a certain configuration of topic/interest dimensions. One evidence that supports this claim is that, although the log likelihood seems to converge, the number of topics (for both models) and interests (only for HDPITM) still slightly fluctuate around a certain value.

From this speculation, we ran both HDP and HDPITM with the different strategy. In particular, we split model training into two periods. In the first period, we allow the model to adjust its configuration, i.e. the dimensions of topics and interests. In the second period, we still train the model but do not allow the dimensions of topics and interests to change. The first one is similar to the training process in the plain HDP and HDPITM. The second one is similar to that of plain LDA and ITM that use the latest configuration from the first period. In this experiment, we set the first period to 500 iterations; another 500 iterations were set for the second phase. Subsequently, ϕ is computed from the last 100 iterations of the second. We refer to this training strategy for HDP as HDP+LDA, and that for HDPITM as HDPITM+ITM. The overall improvement of performance using this strategy are perceived in Figure 8 (c) and (d), comparing to (a) and (b). That is, both HDP+LDA and

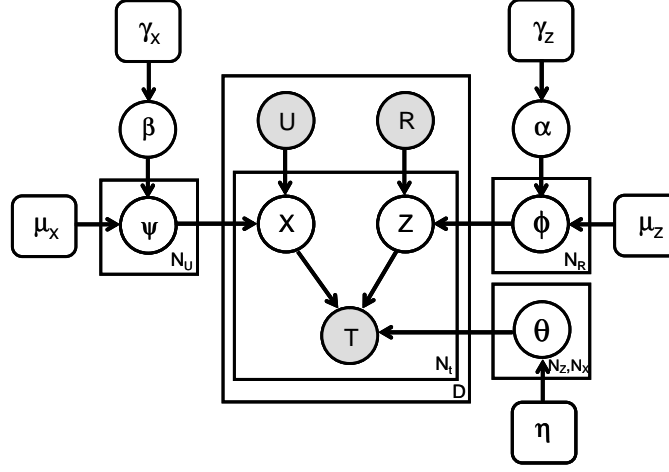


Fig. 7. Graphical representation on the Interest Topic model with hierarchical Dirichlet process (HDPITM).

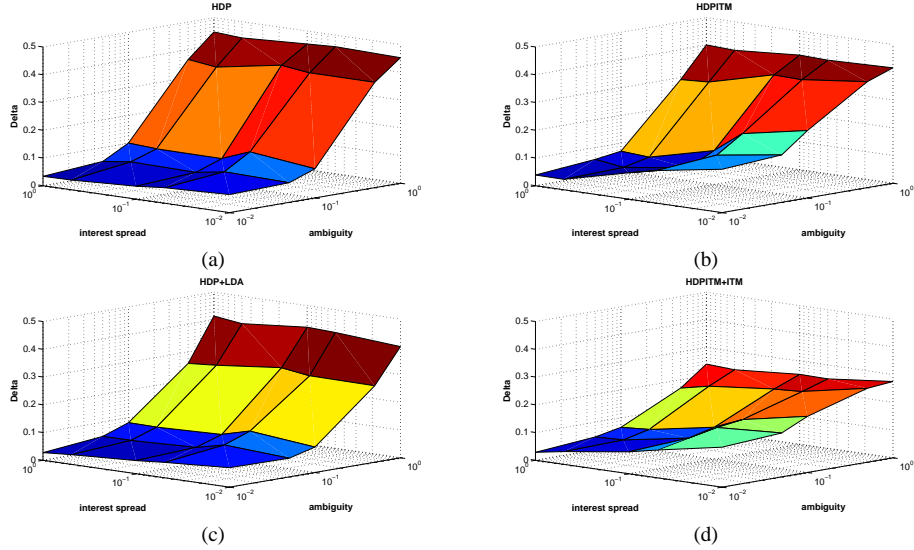


Fig. 8. This plot shows the deviation Δ between actual and learned topics on synthetic data sets, under different degrees of tag-ambiguity and user interest variation. The Δ of HDP is shown on the left (a); as that of HDPITM is on the right (b). As, (c) and (d) shows the deviation produced by HDP+LDA and HDPITM+ITM respectively. For HDP+LDA, new topics can be instantiated, and thus the number of topics can change, during the first half of the run (HDP); then all topics are freed (no new topic can be instantiated) during the second half (LDA). And this is similar to HDPITM+ITM where we take into account user information. See Section 5.1 for more detail.

HDPITM+ITM can produce ϕ , which provide lower Δ , under this strategy. However, HDPITM+ITM performance under the condition with low user interest and low tag ambiguity, is still inferior to HDP+LDA. This is simply because their structures are still the same to those of HDP and HDPITM respectively.

5.2 Performance on the real-world data

In the experiments, we initialize the numbers of topics and interests to 100 and 20 (the number of interests is only applicable to HDPITM), and train the models on the same real-world data sets we used in Section 4.2. The topic and interest assignments are randomly initialized, and then both models are trained with the minimum 400 and maximum 600 iterations. For the first 100 iterations, we allow both models to instantiate a new topic or interest as required, under the constraint that the number of topics and interests does not exceed 400 and 80 respectively. If the model violates this constraint, it will exit this phase early. For the remainder of iterations, we do not allow the model to add new topics or interests (but these numbers can shrink if some topics/interests collapsed during this phase). Then, if the change in log likelihood, averaged over the 10 preceding iterations, is less than 2%, the training process will enter to final learning phase. (See Figure 9 (f) for an example of log likelihood during training iterations.) In fact, we found that the process enters the final phase early in all data sets. In the final phase, consisting of 100 iterations, we use the topic and interest assignments in each iteration to compute the distributions of resources over topics.

The reason we limit the maximum numbers of topics, interests, and iterations over which these models are allowed to instantiate a new topic/interest, is that the numbers of users and tags in our data sets are large, and many new topics and interests could be instantiated. This would require many more iterations to converge, and the models would require more memory than is available on the desktop machine we used in the experiments.¹⁷ We would rather allow the model to “explore” the underlying structure of data within the constraints — in other words, find a configuration which is best suited to the data under a limited exploration period and then fit the data within that configuration. At the end of the parameter estimation, the numbers of allocated topics of HDP models for *flytecomm*, *geocoder*, *wunderground*, *whitepages* and *online-reservationz* was 171, 174, 197, 187 and 175 respectively. The numbers of allocated topics and interests in HDPITM are $\langle 307, 43 \rangle$, $\langle 329, 44 \rangle$, $\langle 231, 81 \rangle$, $\langle 225, 78 \rangle$ and $\langle 207, 72 \rangle$ respectively, which is bigger than those inferred by HDP in all cases. These results suggests that user information allows the HDPITM discover more detailed structure.

HDPITM performs somewhat better than HDP in *flytecomm*, *online-reservationz*, and *geocoder* data sets. Its performance for *wunderground* and *whitepages*, however, is almost identical to HDP. As in Section 4.2, this is possibly due to high interest variation among users. We suspect that weather and directory services are of interest to all users, and are therefore bookmarked by a large variety of users.

6. RELATED RESEARCH

Modeling social annotation is an emerging new field, but it has intellectual roots in two other fields: document modeling and collaborative filtering. It is relevant to the former in that one can view a resource being annotated by users with a set of tags to be analogous to a document, which is composed of words from the document’s authors. Usually, the numbers of users involved in creating a document is much less than those involved in annotating a resource. In regard to collaborative rating systems, annotations created by users in a social annotation system are analogous to object ratings in a recommendation system. However,

¹⁷At maximum, we can only allocate memory for 1,300 Mbytes.

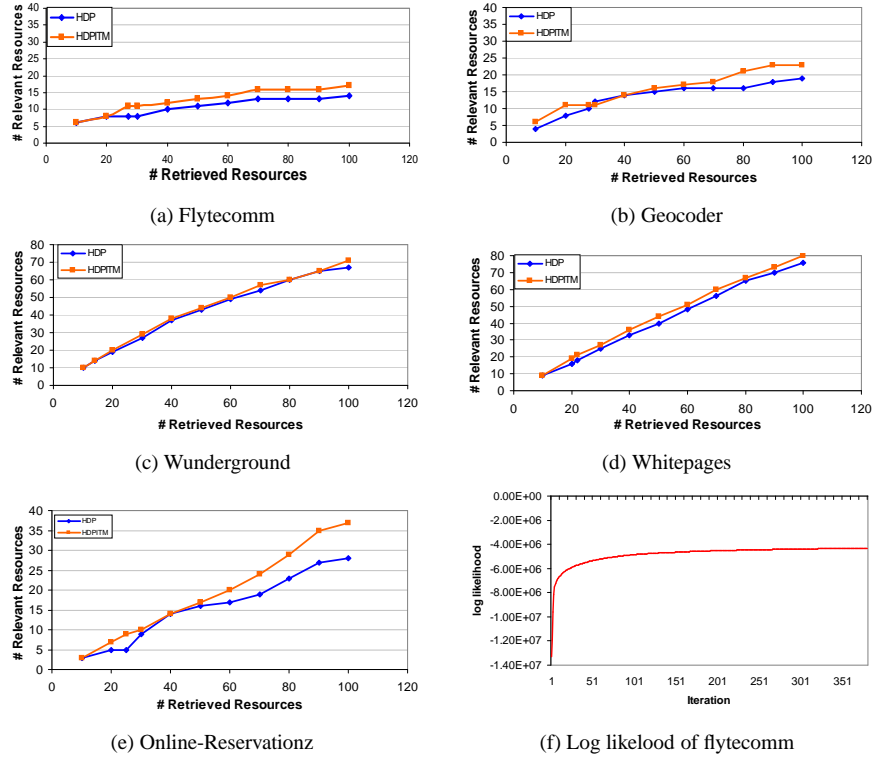


Fig. 9. Performance of different methods on the five data sets (a) – (e). Each plot shows the number of relevant resources (that are similar to the seed) within the top 100 results produced by HDP (non-parametric version of LDA) and HDPITM (nonparametric version of ITM). Each model was initialized with 100 topics and 20 interests for HDPITM. (f) demonstrates log likelihood of the HDPITM model during parameter estimation period of *flytecomm* data set. Similar behavior of the plot (f) is found in both HDP and HDPITM for all data sets.

users only provide one rating to the object in a recommendation system, but they usually annotate an object with several keywords. Therefore, there are several relevant threads of research connecting our work to earlier ones in these areas.

In relation to document modeling, our work is conceptually motivated by the Author-Topic model (AT) [Rosen-Zvi et al. 2004], where we can view a user who annotate a resource as an author who composes a document. In particular, the model explains the process of document generation, governed by author profiles, in forms of distributions of authors over topics. However, this work is not directly applicable to social annotations. This is because, first, in social annotation context, we know who generates a tag on a certain resource; therefore, the author selection process in AT, which selects one of co-authors to be responsible for a generation of a certain document word, is not needed in our context. Second, co-occurrences of user-tag pairs for a certain bookmark are very sparse, i.e., there are fewer than 10 tags per bookmark. Thus, we need to group users who share the same interests together to avoid the sparseness problem. Third, AT has no direct way to estimate distributions of resources over topics since there are only author-topic and topic-word associations in the model. One possible indirect way is to compute this from an

average over all distributions of authors over topics. Our model, instead, explicitly models this distribution, and since it uses profiles of groups of similar users, rather than those of an individual, the distributions are expected to be less biased.

Several recent works apply document modeling to a social annotation. One study [Wu et al. 2006] applies the multi-way aspect model [Hofmann 2001; Popescul et al. 2001] to social annotations on *Delicious*. The model does not explicitly separate user interests and resource topics as our model does, and thus cannot exploit user variations to learn better distributions of resources over topics, as we showed in [Plangprasopchok and Lerman 2007].

[Zhou et al. 2008] introduced a generative model of the process of Web page creation and annotation. The model, called User Content Annotator (UCA), includes words found in Web documents, in addition to tags generated by users to annotate these documents. The authors explore this model in the context of improving IR performance. In this work, a bag of words (tags and content) is generated from two different sources — the document creator and annotator. Although UCA takes documents' contents into account, unlike our model, it makes several assumptions, which we believe do not hold for real-world data. The first assumption is that annotators conceptually agree with the original document's authors (and therefore, share the the same topic space), whereas ITM relaxes this assumption. The second assumption is that users and documents have the same types of distribution over topics, whereas ITM separates *interests* from *topics*. In fact, without documents' content, UCA is almost identical to the Author Topic model [Rosen-Zvi et al. 2004], except for the fact that owners tags are explicitly known, and thus, it shares AT's drawbacks. Another technical drawback of UCA is the following: if a particular tagged Web document has no words (e.g., a Web service, Flickr photo, or YouTube video), UCA would then take into account the taggers only, and lose the variable d that represents the document. Further computation is required to infer $p(z|d)$, the probability of a topic given a document, which is required for the content discovery task we are investigating.

Collaborative filtering was one of the first successful social applications. Collaborative filtering is a technology used by recommender systems to find users with similar interests by asking them to rate items. It then compares their ratings to find users with similar opinions, and recommends to users new items that similar users liked. Among of recent works in collaborative filtering area, [Jin et al. 2006] is most relevant to ours. In particular, the work describe a mixture model for collaborative filtering that takes into account users' intrinsic preferences about items. In this model, item rating is generated from both the item type and user's individual preference for that type. Intuitively, like-minded users would have similar rating on the same item types (e.g., movie genres). When predicting a rating of a certain item for a certain user, the user's previous ratings on other items will be used to infer a like-minded group of users. Then, the "common" rating on that item from the users of that group is the prediction. This collaborative rating process is very similar to that of collaborative tagging. The only technical difference is that each "item" can have multiple "ratings" (in our case, tags) from a single user. This is because an item usually has multiple subjects and each subject can be represented using multiple terms.

There exist, however, major differences between [Jin et al. 2006] and our work. We use the probabilistic model to discover a "resource description" despite users annotating resources with potentially ambiguous tags. Our goal is not to predict how a user will tag a resource (analogous to predicting a rating user will give to an item), or discovering

like-minded groups of users, which our algorithm could also do. The main purpose of our work is to recover the actual “resource description” from noisy observations generated by different users. In essence, we hypothesize that there is actual description of a certain resource and users select and then annotate the resource with that description partially according to their “interest” or “expertise”. In this work, we also demonstrate that when taking into account individual difference in the process, the inferred resource descriptions are not biased toward individual variation as much as those that do not take this issue into account. Another technical difference is that the model is not implemented in fully Bayesian, and uses point estimation to estimate its parameters, which is criticized to be susceptible to local maxima [Griffiths and Steyvers 2004; Steyvers and Griffiths 2006]. Moreover, it can not be extended to allow numbers of topics/interests to be flexible as ours; thus, the strong assumption on the number of topics and interests is required.

Rather than modeling social annotation, [Li et al. 2007] concentrates on an approach that helps users efficiently navigate the Social Web. Although the work share some similar challenges, e.g., tag ambiguity, with ours, the solution proposed in that work is rather different. In particular, the work exploits user activity to resolve ambiguity – as a user selects more tags, the topic scope gets more focused. Consequently, the recently suggested tags associate with fewer and fewer senses, helping to disambiguate the tag. Our approach does not rely on such user activity to disambiguate tag senses; instead, we exploit user interests to do this, since tag sense is correlated with a group of users who share interests. On an applications level, this approach and ours are also different. In particular, the former approach is suitable for situations when users activity and labeled data is available, and can be exploited to filter information on the fly. Our approach, on the other hand, utilizes social annotation only. It is more suitable for batch jobs without user’s intervention; for example, the automatic resource discovery task for mashup applications [Ambite et al. 2009].

7. CONCLUSION

We have presented a probabilistic model of social annotation that takes into account the users who are creating the annotations. We argued that our model is able to learn a more accurate topic description of a corpus of annotated resources by exploiting individual variations in user interests and vocabulary to help disambiguate tags. Our experimental results on collections of annotated Web resources from the social bookmarking site *Delicious* show that our model can effectively exploit social annotation on the resource discovery task.

One issue that our model does not address is tag bias, probably caused by expressiveness of users with high interests in a certain domain. In general, a few users use many more tags than others in annotating resources. This will bias the model toward these users’ annotations, causing the learned topic distributions to deviate from the actual distributions. One possible way to compensate for this is to tie the number of tags to individual interests in the model. ITM also does not at present allow us to include other sources of evidence about documents, e.g., their contents. It would be interesting to extend ITM to include content words, which will make this model more attractive for Information Retrieval tasks.

Since our model is more computationally expensive than other models that ignore user information, e.g. LDA, it is not practical to blindly apply our approach to all data sets. Specifically, our model cannot exploit individual variation in the data that has low tag ambiguity and small individual variation, as shown in Section 4.1. In this case, our model

can only produce small improvement or even similar performance to that of the simpler models. For a practical reason, a heuristic for determining level of tag ambiguity and user variation would be very beneficial in order to determine if the complex model is preferable to the simpler one. Ratios between a number of tags and that of users or that of resources may provide some clues.

As we model the social annotation process by taking into account all essential entities; namely, users, resources and tags, we can apply the model to other applications. For example, one can straightforwardly apply the model to personalize search [Wu et al. 2006; Lerman et al. 2007]. It can also be used to suggest tags to a user annotating a new resource, in the same spirit as rating predictions in Collaborative Filtering.

Appendix

We begin to derive Gibbs sampling equations for ITM in Section 3 from the joint probability of \mathbf{t} , \mathbf{x} and \mathbf{z} of all tuples. Suppose that we have n tuples. Their joint probability is defined as

$$\begin{aligned}
 p(\mathbf{t}, \mathbf{x}, \mathbf{z}; i = 1 : n) &= \int p(\mathbf{t}, \mathbf{x}, \mathbf{z} | \psi, \phi, \theta; i = 1 : n) \cdot p(\psi, \phi, \theta) d\langle \psi, \phi, \theta \rangle \\
 &= c \cdot \int \prod_{i=1:n} (\psi_{u_i, x_i} \cdot \phi_{r_i, z_i} \cdot \theta_{t_i, z_i, x_i}) \cdot \prod_{u, x} \psi_{u, x}^{\beta/N_X - 1} \\
 &\quad \cdot \prod_{r, z} \phi_{r, z}^{\alpha/N_Z - 1} \cdot \prod_{t, x, z} \theta_{t, x, z}^{\eta/N_T - 1} d\langle \psi, \phi, \theta \rangle \\
 &= c \cdot \int \prod_{u, x} \psi_{u, x}^{\sum_i \delta_u(x_i, x) + \beta/N_X - 1} d(\psi) \cdot \int \prod_{r, z} \phi_{r, z}^{\sum_i \delta_r(z_i, z) + \alpha/N_Z - 1} d(\phi) \\
 &\quad \cdot \int \prod_{t, z, x} \theta_{t, z, x}^{\sum_i \delta_{z, x}(t_i, t) + \eta/N_T - 1} d(\theta) \\
 &= c \cdot \prod_r \left(\frac{\prod_z \Gamma(\sum_i \delta_r(z_i, z) + \alpha/N_Z)}{\Gamma(N_r + \alpha)} \right) \cdot \prod_u \left(\frac{\prod_x \Gamma(\sum_i \delta_u(x_i, x) + \beta/N_X)}{\Gamma(N_u + \beta)} \right) \\
 &\quad \cdot \prod_{z, x} \left(\frac{\prod_{t, z, x} \Gamma(\sum_i \delta_{z, x}(t_i, t) + \eta/N_T)}{\Gamma(N_{z, x} + \eta)} \right) \tag{14}
 \end{aligned}$$

where $c = (\frac{\Gamma(\alpha)}{\Gamma(\alpha/N_Z)^z})^r \cdot (\frac{\Gamma(\beta)}{\Gamma(\beta/N_X)^x})^u \cdot (\frac{\Gamma(\eta)}{\Gamma(\eta/N_T)^t})^{(z, x)}$ and $\delta_r(z_i, z)$ is a function which returns 1 if $z_i = z$ and $r_i = r$ otherwise 0. N_r represents a number of all tuples associated with resource r . Similarly, $N_{x, z}$ represents a number of all tuples associated with interest x and topic z .

By rearranging Eq. (14), we obtain

$$\begin{aligned}
 p(\mathbf{t}, \mathbf{x}, \mathbf{z}; i = 1 : n) &= \prod_r \left(\frac{\Gamma(\alpha)}{\Gamma(N_r + \alpha)} \right) \cdot \prod_{r, z} \left(\frac{\Gamma(\sum_i \delta_r(z_i, z) + \alpha/N_Z)}{\Gamma(\alpha/N_Z)} \right) \\
 &\quad \cdot \prod_u \left(\frac{\Gamma(\beta)}{\Gamma(N_u + \beta)} \right) \cdot \prod_{u, x} \left(\frac{\Gamma(\sum_i \delta_u(x_i, x) + \beta/N_X)}{\Gamma(\beta/N_X)} \right)
 \end{aligned}$$

$$\cdot \prod_{x,z} \left(\frac{\Gamma(\eta)}{\Gamma(N_{x,z} + \eta)} \right) \cdot \prod_{x,z,t} \left(\frac{\Gamma(\sum_i \delta_{x,z}(t_i, t) + \eta/N_T)}{\Gamma(\eta/N_T)} \right) \quad (15)$$

Suppose that we have a new tuple and we index this tuple with k (say $k = n + 1$ for convenience). From Eq. (15), we can derive a joint probability of this new tuple k and all other previous tuples as follows

$$\begin{aligned} p (t_k, x_k, z_k, t_i, x_i, z_i; i = 1 : n) \\ &= \frac{\Gamma(\alpha)}{\Gamma(N_{r=r_k} + \alpha + 1)} \cdot \left(\prod_{r \neq r_k} \frac{\Gamma(\alpha)}{\Gamma(N_r + \alpha)} \right) \cdot \frac{\Gamma(\sum_i \delta_{r=r_k}(z_i, z_k) + \alpha/N_Z + 1)}{\Gamma(\alpha/N_Z)} \\ &\cdot \left(\prod_{r \neq r_k, z \neq z_k} \frac{\Gamma(\sum_i \delta_r(z_i, z) + \alpha/N_Z)}{\Gamma(\alpha/N_Z)} \right) \cdot \frac{\Gamma(\beta)}{\Gamma(N_{u=u_k} + \beta + 1)} \cdot \left(\prod_{u \neq u_k} \frac{\Gamma(\beta)}{\Gamma(N_u + \beta)} \right) \\ &\cdot \frac{\Gamma(\sum_i \delta_{u=u_k}(x_i, x_k) + \beta/N_X + 1)}{\Gamma(\beta/N_X)} \cdot \left(\prod_{u \neq u_k, x \neq x_k} \frac{\Gamma(\sum_i \delta_u(x_i, x) + \beta/N_X)}{\Gamma(\beta/N_X)} \right) \\ &\cdot \frac{\Gamma(\eta)}{\Gamma(N_{x=x_k, z=z_k} + \eta + 1)} \cdot \left(\prod_{x \neq x_k, z \neq z_k} \frac{\Gamma(\eta)}{\Gamma(N_{x,z} + \eta)} \right) \\ &\cdot \frac{\Gamma(\sum_i \delta_{x=x_k, z=z_k}(t_i, t_k) + \eta/N_T + 1)}{\Gamma(\eta/N_T)} \cdot \left(\prod_{x \neq x_k, z \neq z_k, t \neq t_k} \frac{\Gamma(\sum_i \delta_{x,z}(t_i, t) + \eta/N_T)}{\Gamma(\eta/N_T)} \right) \end{aligned} \quad (16)$$

For the tuple k , suppose that we only know the values of x_k and t_k while that of z_k is unknown. The joint probability of all tuples, excluding z_k is as follows.

$$\begin{aligned} p (t_k, x_k, t_i, x_i, z_i; i = 1 : n) \\ &= \frac{\Gamma(\alpha)}{\Gamma(N_{r=r_k} + \alpha)} \cdot \left(\prod_{r \neq r_k} \frac{\Gamma(\alpha)}{\Gamma(N_r + \alpha)} \right) \cdot \frac{\Gamma(\sum_i \delta_{r=r_k}(z_i, z) + \alpha/N_Z)}{\Gamma(\alpha/N_Z)} \\ &\cdot \left(\prod_{r \neq r_k, z \neq z_k} \frac{\Gamma(\sum_i \delta_r(z_i, z) + \alpha/N_Z)}{\Gamma(\alpha/N_Z)} \right) \cdot \frac{\Gamma(\beta)}{\Gamma(N_{u=u_k} + \beta + 1)} \cdot \left(\prod_{u \neq u_k} \frac{\Gamma(\beta)}{\Gamma(N_u + \beta)} \right) \\ &\cdot \frac{\Gamma(\sum_i \delta_{u=u_k}(x_i, x_k) + \beta/N_X + 1)}{\Gamma(\beta/N_X)} \cdot \left(\prod_{u \neq u_k, x \neq x_k} \frac{\Gamma(\sum_i \delta_u(x_i, x) + \beta/N_X)}{\Gamma(\beta/N_X)} \right) \\ &\cdot \frac{\Gamma(\eta)}{\Gamma(N_{x=x_k, z=z_k} + \eta)} \cdot \left(\prod_{x \neq x_k, z \neq z_k} \frac{\Gamma(\eta)}{\Gamma(N_{x,z} + \eta)} \right) \cdot \frac{\Gamma(\sum_i \delta_{x=x_k, z=z_k}(t_i, t_k) + \eta/N_T)}{\Gamma(\eta/N_T)} \\ &\cdot \left(\prod_{x \neq x_k, z \neq z_k, t \neq t_k} \frac{\Gamma(\sum_i \delta_{x,z}(t_i, t) + \eta/N_T)}{\Gamma(\eta/N_T)} \right) \end{aligned} \quad (17)$$

By dividing Eq. (15) by Eq. (17), we can obtain the posterior probability of z_k given all other variables as follows

$$\begin{aligned}
p(z_k | t_k, x_k, t_i, x_i, z_i; i = 1 : n) &= \frac{\Gamma(N_{r=r_k} + \alpha)}{\Gamma(N_{r=r_k} + \alpha + 1)} \cdot \frac{\Gamma(\sum_i \delta_{r=r_k}(z_i, z) + \alpha/N_Z + 1)}{\Gamma(\sum_i \delta_{r=r_k}(z_i, z) + \alpha/N_Z)} \\
&\cdot \frac{\Gamma(N_{x=x_k, z=z_k} + \eta)}{\Gamma(N_{x=x_k, z=z_k} + \eta + 1)} \cdot \frac{\Gamma(\sum_i \delta_{x=x_k, z=z_k}(t_i, t_k) + \eta/N_T + 1)}{\Gamma(\sum_i \delta_{x=x_k, z=z_k}(t_i, t_k) + \eta/N_T)} \\
&= \frac{\sum_i \delta_{r=r_k}(z_i, z_k) + \alpha/N_Z}{N_{r=r_k} + \alpha} \cdot \frac{\sum_i \delta_{x=x_k, z=z_k}(t_i, t_k) + \eta/N_T}{N_{x=x_k, z=z_k} + \eta} \\
&= \frac{N_{r=r_k, z=z_k} + \alpha/N_Z}{N_{r=r_k} + \alpha} \cdot \frac{N_{x=x_k, z=z_k, t=t_k} + \eta/N_T}{N_{x=x_k, z=z_k} + \eta} \tag{18}
\end{aligned}$$

Intuitively, we can perceive from Eq. (18) that $\frac{N_{r=r_k, z=z_k} + \alpha/N_Z}{N_{r=r_k} + \alpha}$ tell us how resource r is likely to be described by the topic z ; as the later part, $\frac{N_{x=x_k, z=z_k, t=t_k} + \eta/N_T}{N_{x=x_k, z=z_k} + \eta}$ tell us how tag t is likely to be chosen given interest x and z .

Similarly, we can obtain the posterior probability of x_k as we did for z_k .

$$\begin{aligned}
p(x_k | t_k, z_k, t_i, x_i, z_i; i = 1 : n) &= \frac{N_{u=u_k, x=x_k} + \beta/N_X}{N_{u=u_k} + \beta} \\
&\cdot \frac{N_{x=x_k, z=z_k, t=t_k} + \eta/N_T}{N_{x=x_k, z=z_k} + \eta} \tag{19}
\end{aligned}$$

We can now generalize Eq. (18) and Eq. (19) for sampling posterior probabilities of topic z and interest x of a present tuple i given all other tuples. We define $N_{r_i, z_{-i}}$ as the number of all toples having $r = r_i$ and z but excluding the present tuple i . Similarly, N_{z_{-i}, x_i, t_i} is a number of all tuples having $x = x_i$, $t = t_i$ and z but excluding the present tuple i . As \mathbf{z}_{-i} represents all topic assignments except that of the tuple i .

$$p(z_i | \mathbf{z}_{-i}, \mathbf{x}, \mathbf{t}) = \frac{N_{r_i, z_{-i}} + \alpha/N_Z}{N_{r_i} + \alpha - 1} \cdot \frac{N_{z_{-i}, x_i, t_i} + \eta/N_T}{N_{z_{-i}, x_i} + \eta} \tag{20}$$

$$p(x_i | \mathbf{x}_{-i}, \mathbf{z}, \mathbf{t}) = \frac{N_{u_i, x_{-i}} + \beta/N_X}{N_{u_i} + \beta - 1} \cdot \frac{N_{x_{-i}, z_i, t_i} + \eta/N_T}{N_{x_{-i}, z_i} + \eta} \tag{21}$$

ACKNOWLEDGMENTS

We would like to thank anonymous reviewers for providing useful comments and suggestions to improve the manuscript. This material is based in part upon work supported by the National Science Foundation under Grant Numbers CMMI-0753124 and IIS-0812677. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

- AMBITE, J. L., DARBHA, S., GOEL, A., KNOBLOCK, C. A., LERMAN, K., PARUNDEKAR, R., AND RUSS, T. A. 2009. Automatically constructing semantic web services from online sources. In *Proceedings of International Semantic Web Conference*. 17–32.

- BLEI, D. M., NG, A. Y., AND JORDAN, M. I. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3, 993–1022.
- BUNTINE, W., PERTTU, S., AND TUULOS, V. 2004. Using discrete pca on web pages. In *Proceedings of ECML workshop on Statistical Approaches to Web Mining*.
- BUNTINE, W. L. 1994. Operations for learning with graphical models. *Journal of Artificial Intelligence Research* 2, 159–225.
- ESCOBAR, M. D. AND WEST, M. 1995. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association* 90, 577–588.
- GILKS, W., RICHARDSON, S., AND SPIEGELHALTER, D. 1996. *Markov Chain Monte Carlo in Practice*. Interdisciplinary Statistics. Chapman & Hall.
- GOLDER, S. A. AND HUBERMAN, B. A. 2006. Usage patterns of collaborative tagging systems. *Journal of Information Science* 32, 2 (April), 198–208.
- GRIFFITHS, T. L. AND STEYVERS, M. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America* 101, 5228–5235.
- HOFMANN, T. 1999. Probabilistic latent semantic analysis. In *Proceedings of 15th Conference on Uncertainty in Artificial Intelligence*. 289–296.
- HOFMANN, T. 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning* 42, 1-2, 177–196.
- JIN, R., SI, L., AND ZHAI, C. 2006. A study of mixture models for collaborative filtering. *Information Retrieval* 9, 3, 357–382.
- LERMAN, K., PLANGPRASOPCHOK, A., AND WONG, C. 2007. Personalizing image search results on flickr. In *Proceedings of AAAI workshop on Intelligent Web Personalization*.
- LI, R., BAO, S., YU, Y., FEI, B., AND SU, Z. 2007. Towards effective browsing of large scale social annotations. In *Proceedings of the 16th international conference on World Wide Web*. 943–952.
- LIN, J. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory* 37, 1, 145–151.
- MARLIN, B. 2004. Collaborative filtering: A machine learning perspective. M.S. thesis, University of Toronto, Toronto, Ontario, Canada.
- MCCALLUM, A., WANG, X., AND CORRADA-EMMANUEL, A. 2007. Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research* 30, 249–272.
- MIKA, P. 2007. Ontologies are us: A unified model of social networks and semantics. *Web Semantics* 5, 1, 5–15.
- MINKA, T. P. 2001. Expectation propagation for approximate bayesian inference. In *Proceedings of the 17th conference on Uncertainty in Artificial Intelligence*. 362–369.
- NEAL, R. M. 2000. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics* 9, 2, 249–265.
- PLANGPRASOPCHOK, A. AND LERMAN, K. 2007. Exploiting social annotation for automatic resource discovery. In *Proceedings of AAAI workshop on Information Integration on the Web*.
- POPESCU, A., UNGAR, L., PENNOCK, D., AND LAWRENCE, S. 2001. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of 17th Conference on Uncertainty in Artificial Intelligence*. 437–444.
- RASMUSSEN, C. E. 2000. The infinite gaussian mixture model. In *Proceedings of Advances in Neural Information Processing Systems 12*. 554–560.
- RATTENBURY, T., GOOD, N., AND NAAMAN, M. 2007. Towards automatic extraction of event and place semantics from flickr tags. In *Proceedings of the 30th Annual ACM SIGIR Conference on Research and Development in Information Retrieval*. 103–110.
- RITTER, C. AND TANNER, M. A. 1992. Facilitating the gibbs sampler: The gibbs stopper and the griddy-gibbs sampler. *Journal of the American Statistical Association* 87, 419, 861–868.
- ROSEN-ZVI, M., GRIFFITHS, T., STEYVERS, M., AND SMYTH, P. 2004. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in Artificial Intelligence*. 487–494.
- SAHU, S. K. AND ROBERTS, G. O. 1999. On convergence of the em algorithm and the gibbs sampler. *Statistics and Computing* 9, 9–55.
- SCHMITZ, P. 2006. Inducing ontology from flickr tags. In *Proceedings of WWW workshop on Collaborative Web Tagging*.

- STEYVERS, M. AND GRIFFITHS, T. 2006. Probabilistic topic models. In *Latent Semantic Analysis: A Road to Meaning*, T. Landauer, D. Mcnamar, S. Dennis, and W. Kintsch, Eds.
- TEH, Y. W., JORDAN, M. I., BEAL, M. J., AND BLEI, D. M. 2006. Hierarchical dirichlet processes. *Journal of the American Statistical Association* 101, 1566–1581.
- WU, X., ZHANG, L., AND YU, Y. 2006. Exploring social annotations for the semantic web. In *Proceedings of the 15th international conference on World Wide Web*. 417–426.
- ZHOU, D., BIAN, J., ZHENG, S., ZHA, H., AND GILES, C. L. 2008. Exploring social annotations for information retrieval. In *Proceedings of the 17th international conference on World Wide Web*. 715–724.